Ein erweiterbares System für die Spezifikation und Generierung interaktiver Selbsttestaufgaben

Manfred Widera[1], Barbara Messing[1], Gabriele Kern-Isberner[2], Malte Isberner[1], Christoph Beierle[1]

[1]Fachbereich Informatik, Wissensbasierte Systeme,
FernUniversität in Hagen, 58084 Hagen
[2]Fachbereich Informatik, Information Engineering,
Universität Dortmund, 44221 Dortmund

Zusammenfassung

Um im Rahmen eines Fernstudiums die direkte Rückmeldung über den eigenen Lernerfolg zu erleichtern, wurde das ASTERIX-System entwickelt, auf das mittels einer Internet-Schnittstelle zugegriffen werden kann. ASTERIX enthält interaktive Selbsttestaufgaben unterschiedichen Typs aus dem Bereich der formalen Grundlagen der Informatik, deren Implementierung automatisch aus XML-Spezifikationen generiert wird. In dieser Arbeit stellen wir die wesentlichen Funktionen des Systems und die von ihm unterstützen Aufgabentypen vor und zeigen auf, wie ASTERIX im Rahmen der Lehre an der FernUniversität in Hagen eingesetzt wird.

1 Einleitung

Das in dieser Arbeit beschriebene System ASTERIX [BIKI+05] ist ein interaktives Angebot für die Informatik-Ausbildung an der FernUniversität Hagen, das die normalen Studienveranstaltungen ergänzt. Um zu verdeutlichen, in welchem Kontext dieses System eingesetzt wird, geben wir zuvor eine kurze Beschreibung des Ablaufs einer vorlesungsäquivalenten Lehrveranstaltung an der FernUniversität.

Bausteine der Informatik-Ausbildung an der FernUniversität Hagen sind Kurstexte und Einsendeaufgaben, die in regelmäßigen Abständen an die Kursteilnehmerinnen und Kursteilnehmer verschickt werden. Die Kurstexte enthalten den für das Selbststudium aufbereiteten Vorlesungsstoff; die Einsendeaufgaben haben den Sinn, die Studenten zu einer kontinuierlichen Beschäftigung mit den Kursinhalten zu motivieren und mit Abgabefristen einen zeitlichen Rahmen für die Bearbeitung zu setzen. Um das Verständnis schon während der Bearbeitung des Kursmaterials

selbst kontrollieren zu können, sind in den Kurstext sogenannte Selbsttestaufgaben integriert; zu diesen sind am Ende der jeweiligen Kurseinheit Lösungsvorschläge angegeben. Die Korrektur der Einsendeaufgaben gibt den Teilnehmern eine individuelle Rückmeldung über ihren Lernerfolg. Diese Rückmeldung kann schon aus organisatorischen Gründen nur zeitlich versetzt erfolgen; gerade bei großen Veranstaltungen wie den obligatorischen Anfängerkursen nimmt die Korrektur einige Zeit in Anspruch, währenddessen die Bearbeitung des Kurses fortschreitet. Das Feedback über die Einsendeaufgaben hat sich bewährt; um die Nachteile der zeitlichen Verschleppung auszugleichen, sind jedoch, besonders für die ersten Semester, weitere Betreuungsangebote notwendig. Hier sind insbesondere die betreuten Newsgroups bzw. Internet-Foren zu nennen, die vor allem von den Teilnehmern der Anfängerkurse oft benutzt werden, um Schwierigkeiten zu besprechen und sich auszutauschen.

Im Vergleich zu Studentinnen und Studenten an der Präsenzuniversität sind die Voraussetzungen, die Teilnehmer an Fernuniversitätskursen mitbringen, besonders heterogen. Ein großer Teil der Fernstudierenden hat bereits eine abgeschlossene Berufsausbildung und studiert nebenberuflich, nicht selten mit äußerst knappen zeitlichen Ressourcen. Einige hatten in ihrer bisherigen Laufbahn wenig Berührungspunkte mit mathematischen Methoden, wie sie in der Informatik zwingend gebraucht werden. Schon Basisfertigkeiten wie das Bilden einer Potenzmenge können dann zum Problem werden. Diese Schwierigkeiten lassen sich nur bewältigen, wenn die erlernten Methoden immer wieder geübt werden. Wenn aber der persönliche Kontakt so reduziert ist, wie es bei einem Fernstudium nun einmal typisch ist, fehlt die direkte Lernkontrolle. Dadurch bleibt die Frage „Ist das so richtig?“ allzu häufig offen.

Das mit Asterix entwickelte tutorielle Programm ergänzt die Selbsttestaufgaben aus den Kurstexten durch ein interaktives Übungsprogramm, auf das die Teilnehmer über das Internet zugreifen können. Dieses System wurde als begleitendes Angebot zum Kurs *Formale Grundlagen der Informatik* entwickelt, einem Kurs, der für das erste Semester im Bachelor-Studiengang Informatik Pflicht ist. Vermittelt werden elementare Kenntnisse unter anderem über Mengenlehre, Relationen/Funktionen und Logik. Asterix bietet eine zusätzliche Übungsmöglichkeit mit automatischer und direkter Rückmeldung. Hilfestellungen und Hinweise zu Fehlern sind integriert.

Auch abgesehen davon, dass der Aufwand für dieses Angebot angesichts hoher Teilnehmerzahlen lohnend erscheint, ist der Stoff dieser Veranstaltung für ein Online-Übungsangebot besonders geeignet. Beim Schneiden zweier Mengen oder dem Negieren einer prädikatenlogischen Formel gibt es nun einmal wenig Ermessensspielraum. Schon einfache Ja/Nein-Fragen können Verständnislücken bloßlegen. Zudem gibt es einen reichen Fundus an Beispielen und Aufgaben. Hinweise zu häufig auftretenden Fehlern rekrutieren sich meist aus Erfahrungen, die in früheren Semestern mit diesen Aufgaben gemacht wurden, und Fragen, die in den Newsgroups aufgetaucht sind. Nützlich ist das Übungsangebot auch deshalb, weil die mathematischen und formalen Basiskenntnisse, die in dieser Veranstal-

tung vermittelt werden, aus Teilnehmersicht recht weit von Informatikanwendungen entfernt sind. Da ist es schön zu sehen, dass wenigstens ein kleiner Teil nicht nur auf dem Papier stattfindet. Jeder Anreiz, sich mit dem Stoff zu beschäftigen, bringt Vorteile für den erfolgreichen Verlauf des Studiums, denn nur wer die formalen Methoden sicher anwenden kann, ist auch in der Lage, fortgeschrittene Techniken zu erlernen.

Neben einfachen Multiple-Choice-Aufgaben bietet ASTERIX eine Reihe unterschiedlicher Aufgabentypen an, z.B. Zuordnungsaufgaben, Aufgaben mit Wahrheitswertetabellen zur Evaluation aussagenlogischer Formeln oder Aufgaben, bei denen Mengen bestimmt und eingegeben werden müssen. Um einen größtmöglichen Grad an Portabilität und Erweiterbarkeit zu erzielen, erfolgt die interne Darstellung der Aufgaben systemunabhängig in XML. Aus der XML-Darstellung generiert ein Parser Code für die interaktive Darstellung, Bearbeitung und Korrektur der Aufgaben, auf die über das Internet zugegriffen werden kann. Über eine Schnittstelle wird ASTERIX an den Lernraum Virtuelle Universität (LVU) der FernUniversität Hagen [LVU05] integriert.

Im folgenden Abschnitt 2 wird die Architektur des Systems ASTERIX und seine Implementierung beschrieben. In Abschnitt 3 werden einige Beispielaufgaben gezeigt, die die wesentlichen Funktionen des Systems demonstrieren. Abschnitt 4 stellt die von ASTERIX unterstützten Aufgabentypen vor. Damit wird ein Schema für die Entwicklung neuer Aufgaben gegeben. In Abschnitt 5 werden die XML-Spezifikationen zu den Beispielaufgaben skizziert. Die Arbeit schließt mit Zusammenfassung und Ausblick in Abschnitt 6.

2 Systemarchitektur und Implementierung

ASTERIX besteht aus mehreren Modulen. Ein Parser übernimmt die Aufbereitung des XML-Aufgabencodes für die interaktive Bearbeitung, deren Ablauf vom Aufgabenmodul gesteuert wird. Ein Korrekturmodul überprüft die Eingaben des Benutzers auf ihre Korrektheit hin. In Abhängigkeit davon, ob eine (Teil-)Aufgabe richtig oder falsch gelöst wurde, generiert das Kommentarmodul Anmerkungen oder Fehlermeldungen bzw. Tipps, so dass der Benutzer nicht nur die Richtigkeit seiner Lösungen direkt überprüfen kann, sondern auch noch relevante Informationen zur Aufgabe im Themenzusammenhang bekommt. Abbildung 1 visualisiert das Zusammenspiel der einzelnen Module; innerhalb des Lernraums Virtuelle Universität der FernUniversität in Hagen [LVU05] soll die Schnittstelle des virtuellen Informatiklabors [LGH02] für die Internetanbindung genutzt werden.

ASTERIX wurde in C++ implementiert und verwendet für eine Reihe von Standardaufgaben (z.B. Darstellung der Benutzeroberfläche, Parsing der XML-Dateien) die *Qt*-Bibliothek der Firma *Trolltech*. Als Plattform wurde Linux verwendet, ASTERIX lässt sich jedoch auf allen Plattformen, auf denen diese Bibliothek verfügbar ist, ausführen. Für die Kompilation steht ein Makefile bereit. Die Gesamtzeilenanzahl aller (nicht automatisch generierten) Quelldateien beträgt

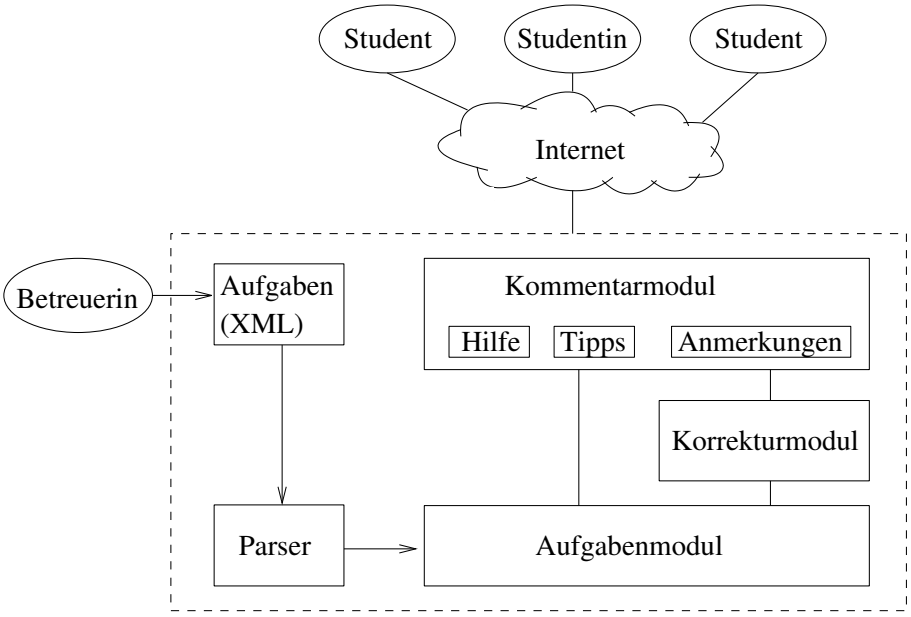

Abb. 1: Das ASTERIX-System

ca. 6600.

ASTERIX ist objektorientiert aufgebaut. Für jeden einzelnen Aufgabentyp gibt es eine Klasse, die sich von einer Basisklasse für Aufgabentypen ableitet. Diese Klassen erhalten die XML-Daten der jeweiligen Aufgabe vom XML-Parser, bauen daraus die Benutzeroberfläche auf und übernehmen die Auswertung der Benutzereingaben. Die Erstellung der Aufgabenklasse aus der Basisklasse erfolgt dynamisch in Abhängigkeit vom Aufgabentyp, der vom Parser eingelesen wird. Dadurch ist es möglich, mit nur einer kleinen Änderung an den bestehenden Quelldateien einen neuen Aufgabentyp hinzuzufügen. Weiterhin ist es leicht möglich, neue Aufgabentypen über dynamisch geladenene Bibliotheken (unter Windows DLLs, unter Linux *.so*s) zu implementieren und damit das System an individuelle Aufgabenstellungen anzupassen.

3 Aufgabenbearbeitung in ASTERIX

Aus Studentensicht präsentiert das System ASTERIX nach erfolgter Anmeldung und Auswahl einer Aufgabe eine Aufgabendarstellung, die die zu lösende Aufgabe beschreibt und angibt, in welchem Format die Lösung einzugeben ist. Die Eingabefelder und -formate sind vom Aufgabentyp abhängig. Weiterhin kann jede Aufgabe aus mehreren Teilaufgaben bestehen, die sich durch Auswahl der entsprechenden Reiter ansteuern lassen.

Die Auswertung der Eingabe wird durch Betätigung des Buttons *Auswerten* gestartet. Die Aufgabenbearbeitung ist ein iteraktiver Prozess, da bereits Teilein-

gaben von ASTERIX ausgewertet werden können und der Student anschließend weitere Eingaben machen kann. Stellt ASTERIX bei der Auswertung Fehler in den eingegebenen Antworten fest, hat der Student die Möglichkeit, im nächsten Schritt diese Fehler zu korrigieren, wobei es in Abhängigkeit vom Aufgabentyp sein kann, dass z.B. falsch angekreuzte Lösungen nicht noch einmal zur Auswahl angeboten werden.

Zu jeder Aufgabe kann es Hilfestellungen, Tipps und Anmerkungen geben:

- *Hilfe* lässt sich vor Bearbeitung der Aufgabe anklicken, so dass der Verweis auf den Hilfetext zusammen mit der Aufgabe sichtbar ist.
- *Tipp* wird angezeigt, wenn eine (Teil-)Aufgabe falsch gelöst wurde. Im Gegensatz zu einer *Fehler*-Anzeige handelt es sich um einen aufgabenspezifischen Text, der zusätzlich erklärt, *wieso* die Lösung des Benutzers falsch ist.
- *Anmerkung* wird angezeigt, nachdem die Aufgabe korrekt gelöst wurde.

Im Folgenden geben wir einige Aufgaben aus verschiedenen Themenbereichen und Beispiele unterschiedlicher Aufgabentypen an, die von ASTERIX unterstützt werden.

Abbildung 2 zeigt eine Aufgabe zum Thema Mengen und Prädikatenlogik. Ausdrücke der Mengensprache müssen prädikatenlogischen Formeln zugeordnet werden. Die Zuordnung erfolgt über das Anklicken der entsprechenden Buttons auf der linken und der rechten Seite. Neben drei korrekten Zuordnungen ist in Abb. 2 eine falsche Zuordnung erfolgt, die in einer entsprechenden Fehlermeldung resultiert.

Das Arbeiten mit Wahrheitstabellen spielt in der Aussagenlogik eine große Rolle. Abbildung 3 zeigt, wie mit Hilfe einer interaktiven Wahrheitstabelle der Wert einer aussagenlogischen Formel bestimmt werden kann.

Will man bei Aufgaben aus dem Bereich der Mengenlehre nicht schon Antwortmöglichkeiten vorgeben, so ist es erforderlich, die direkte Eingabe von Mengen und Mengenausdrücken zu unterstützen. In der Aufgabe in Abb. 4 sollen zu gegebenen Grundmengen Mengenausdrücke ausgewertet werden. Die Eingabe erfolgt in der üblichen mathematischen Mengenschreibweise.

Bei der in Abb. 5 dargestellten Aufgabe geht es darum, grundlegende Eigenschaften von Relationen einzuüben. Zu gegebenen verschiedenen Relationen (z.B. *X kennt Y*) ist hier jeweils anzugeben, ob die Eigenschaften der Symmetrie, Reflexivität und Transitivität erfüllt sind. Die Anwort erfolgt über Auswahlfelder, wobei jeweils mehrere Antworten korrekt sein können (X aus N-Auswahl). Der in Abb. 5 gezeigte Tipp gibt ein anschauliches Gegenbeispiel, warum die Relation in der zweiten Zeile (*X und Y belegen einen gemeinsamen Kurs*) nicht — wie hier von dem Benutzer angegeben — transitiv ist.

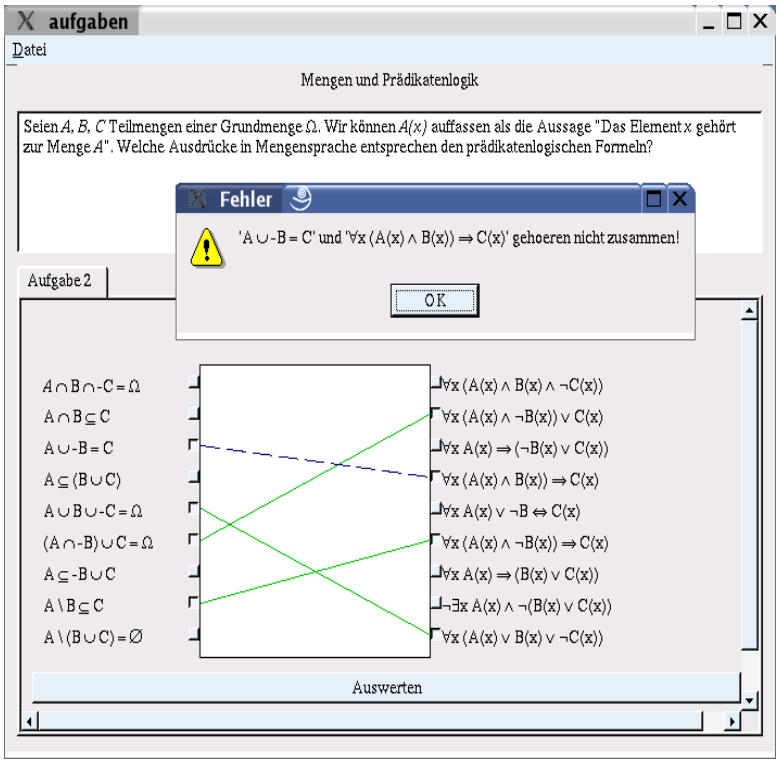

Abb. 2: Zuordnungsaufgabe zum Thema Mengen und Prädikatenlogik

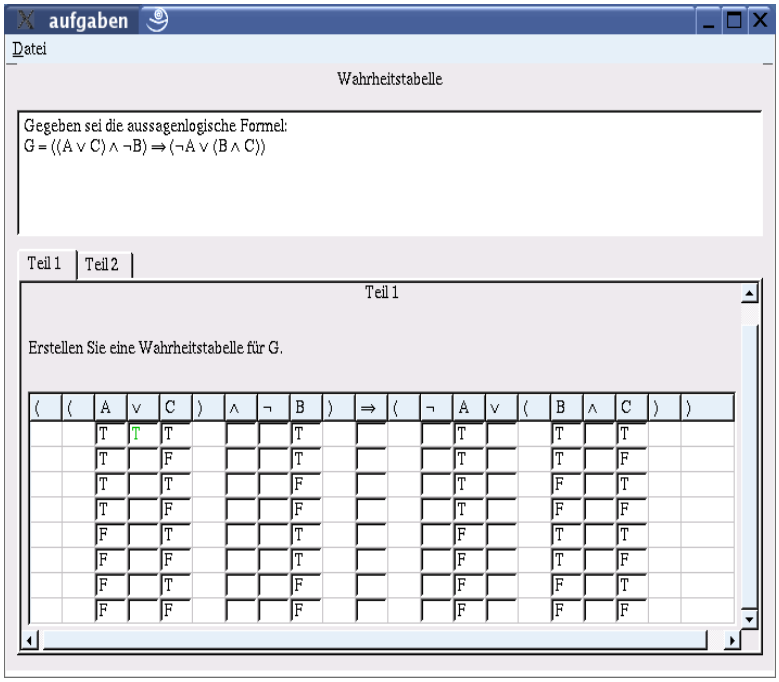

Abb. 3: Interaktive Wahrheitstabelle für eine aussagenlogische Formel

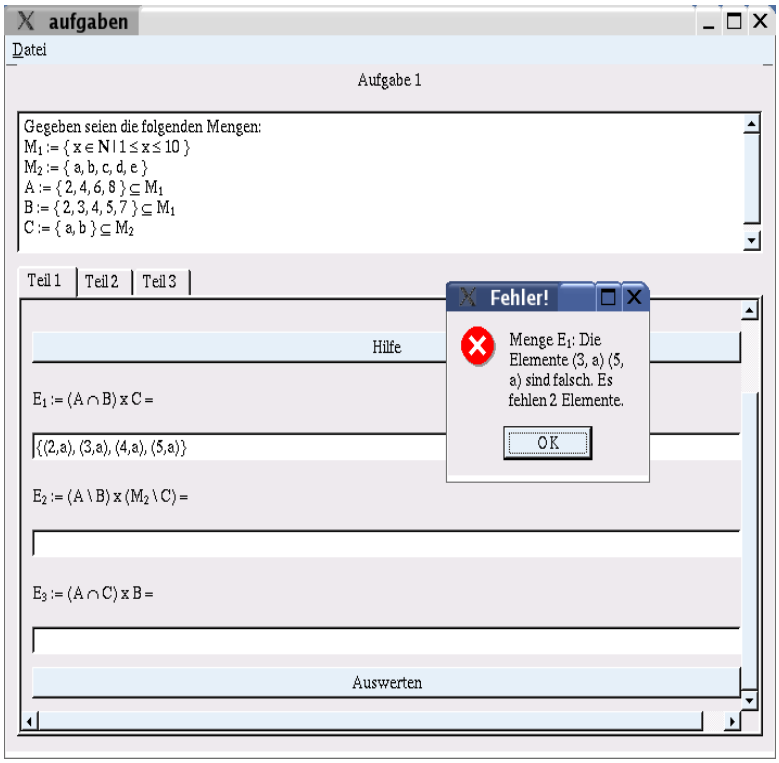

Abb. 4: Berechnung von Mengen mit direkter Eingabe in Mengenschreibweise

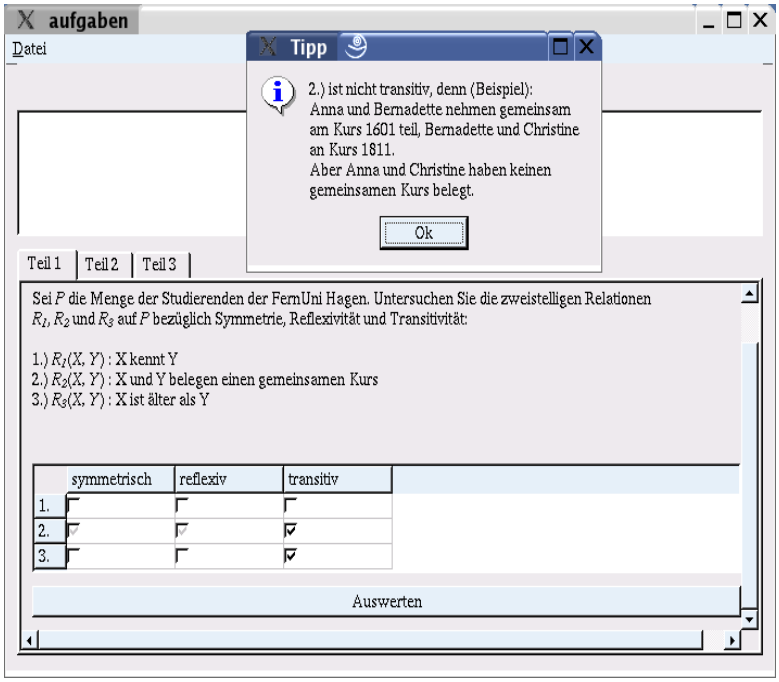

Abb. 5: X aus N-Tabelle zur Einübung elementarer Eigenschaften von Relationen

4 Unterstützte Aufgabentypen

Asterix unterstützt eine Reihe unterschiedlicher Aufgabentypen, die zum Teil deutlich über die Funktionalität einfacher Multiple Choice-Aufgaben hinaus gehen:

Bei der **1 aus N**-Auswahl werden zur Aufgabenstellung mehrere (N) potenzielle Lösungen angegeben, von denen genau eine richtig ist. Der Student wählt genau eine dieser Lösungen aus. Ist diese falsch, wird bei seiner Korrektur diese Lösungsmöglichkeit nicht erneut zur Auswahl angeboten.

Bei der **X aus N**-Auswahl werden zu der Aufgabenstellung mehrere (N) potenziell richtige Lösungen angegeben, von denen eine unbestimmte Anzahl (X) richtig ist. Der Student kann beliebig viele dieser Lösungen markieren. Nach Auswertung sind die falsch ausgewählten bei seiner Korrektur nicht mehr selektierbar, die richtigen nicht mehr deselektierbar.

Beim Aufgabentyp **Wahrheitstabelle** gibt die Betreuerin eine aussagenlogische Formel an, die der Student auswerten soll. Dem Studenten wird eine Tabelle mit den möglichen Belegungen der in der Formel auftretenden Variablen präsentiert. Für jeden Teilausdruck gibt es ein Feld, in dem er den Wert dieses Teilausdrucks unter der betreffenden Belegung eintragen kann (vgl. Abb. 3).

Bei der **1 aus N-Tabelle** werden mehrere Aufgaben des Typs 1 aus N-Auswahl, für die die Mengen der wählbaren Antworten übereinstimmen, in einer Tabelle zusammengefasst. Dabei stehen die Spalten der Tabelle für die verschiedenen Antworten, die Zeilen geben die Fragen an. In jeder Zeile befinden sich Auswahlknöpfe, von denen jeweils nur einer auswählbar ist. Am Rand der Tabelle sieht der Benutzer für jede Zeile, ob seine Antwort richtig war.

Die **X aus N-Tabelle** ist ähnlich der 1 aus N-Tabelle, abgesehen davon, dass in einer Zeile wie bei Aufgaben des Typs X aus N-Auswahl auch mehrere Antworten ausgewählt werden können (vgl. Abb. 5).

Bei der **Texteingabe** beantwortet der Student die in der Aufgabe gestellten Fragen durch textuelle oder numerische Eingabe in dafür vorgesehene Felder. Bei Bedarf können dies auch mathematische Ausdrücke sein, so dass z.B. statt eines numerischen Eintrags wie “256” alternativ auch “2^8” eingegeben werden kann. Für die automatische Korrektur werden die Antworten mit den korrekten Lösungen, die in der internen Repräsentation der Aufgabe von der Betreuerin spezifiziert worden sind, verglichen.

Der Aufgabentyp **Auswahl** arbeitet prinzipiell wie die 1 aus N-Tabelle, allerdings erfolgt die Darstellung nicht in Tabellenform, sondern in Form von Dropdown-Listen, die direkt nach der jeweiligen Frage erscheinen.

Der Aufgabentyp **Mengenangabe** erlaubt ein komplexes Eingabeformat für die Angabe von Mengen: Als Elemente können von den Studenten Zahlen, Buch-

staben, Tupel und Untermengen angegeben werden. Die in den Aufgabenstellungen spezifizierten Mengenausdrücke können darüber hinaus Operationen auf Mengen wie Durchschnitt, Vereinigung oder kartesisches Produkt enthalten. Die korrekten Ergebnismengen werden automatisch aus diesen vorgegeben Mengenausdrücken berechnet und mit den von den Studenten eingegebenen Mengen verglichen (vgl. Abb. 4).

Beim **Zuordnung**-Aufgabentyp muss der Student jeweils zwei zusammengehörige Elemente aus zwei verschiedenen Listen einander zuordnen, wobei die Verbindungen grafisch dargestellt werden (vgl. Abb. 2). Die Reihenfolge der Elemente auf der linken Seite wird aus der internen Aufgabenspezifikation in XML (vgl. Abschnitt 5) übernommen, die der Elemente auf der rechten Seite wird jedesmal neu (zufällig) generiert.

5 XML-Spezifikationen

Um einen größtmöglichen Grad an Portabilität und Erweiterbarkeit zu erzielen, erfolgt die interne Repräsentation der Aufgaben systemunabhängig in XML. XML ist anerkannter Standard für die interne Verwaltung von Lehrinhalten [BCH+03] und wird in vielen Projektvorhaben in diesem Bereich eingesetzt (z.B. [WWR05]). Die Argumentation für XML lässt sich direkt auf ASTERIX übertragen. Für die Aufgaben findet die XML-Version 1.0 [BPSM+04] Verwendung.

5.1 Allgemeine Angaben

Das Toplevel-Element jeder Aufgaben-Datei ist der Tag `aufgabenblatt`. Alle zu einer Aufgabe gehörenden Teilaufgaben stehen in diesem Tag, außerdem eine Überschrift und ein beschreibender Text. Ein `aufgabe`-Tag beginnt eine neue Teilaufgabe und ist ähnlich dem `aufgabenblatt`-Tag aufgebaut. Ein Attribut `typ` gibt den Typ der Aufgabe an. Weitere allgemeine Tags sind beispielsweise `help` oder `tipp`, mit denen die entsprechenden Hinweistexte markiert werden. Darüber hinaus gibt es für jeden Aufgabentyp spezifische Tags und Attribute. Im Folgenden gehen wir zur Illustration auf die XML-Spezifikation der in Abschnitt 3 vorgestellten Beispielaufgaben ein.

5.2 Aufgabentyp Zuordnung

Der `typ`-Wert einer *Zuordnungs*-Aufgabe ist `zuordnen`. Der typspezifische Tag `antwort` legt ein Antwortpaar fest. Die Attribute `left` und `right` bestimmen die beiden zueinander gehörenden Einträge auf der linken bzw. der rechten Seite. Bei der Generierung der grafischen Aufgabendarstellung, bei der die Elemente der linken und rechten Seite durch Linien verbunden werden müssen, wird die Reihenfolge der Elemente auf der rechten Seite jedesmal in einer zufälligen Weise

neu festgelegt. In Abb. 6 ist die XML-Spezifikation der Aufgabe aus Abb. 2 angegeben.

```
<?xml version="1.0"?>
<aufgabenblatt>
<titel>Mengen und Prädikatenlogik</titel>
<text>Seien [i]A, B, C[/i] Teilmengen einer Grundmenge \{omega}.
      Wir können [i]A(x)[/i] auffassen als die Aussage
      "Das Element [i]x[/i] gehört zur Menge [i]A[/i]".
      Welche Ausdrücke in Mengensprache entsprechen den
      prädikatenlogischen Formeln?</text>
<aufgabe typ="zuordnen">
<titel>Aufgabe 2</titel>
<help>Beispielsweise werden mit A \{intersect} -B diejenigen
      Elemente von \{omega} beschrieben, für die A
      zutrifft, B jedoch nicht. Beachten Sie, dass auf
      der linken Seite Aussagen über Beziehungen von
      Mengen (Teilmengenbeziehung, Gleichheit) stehen.
      Formulieren Sie diese erst in Umgangssprache, dann
      in Prädikatenlogik. A \{intersect} -B = A heißt
      beispielsweise: A geschnitten mit dem Komplement von
      B ergibt wieder A, d.h. die Elemente von A \{intersect}
      -B und A sind gleich, als PL-Formel: \{forall}[i]x[/i]
      (A([i]x[/i]) \{and} \{not}B([i]x[/i])) \{coimpl}
      A([i]x[/i]).</help>
<antwort left="A \{intersect} B \{intersect} -C = \{omega}"
         right="\{forall}x (A(x) \{and} B(x) \{and} \{not}C(x))"/>
<antwort left="A \{intersect} B \{subseteq} C"
         right="\{forall}x (A(x) \{and} B(x)) \{impl} C(x)"/>
<antwort left="A \{union} -B = C"
         right="\{forall}x A(x) \{or} \{not}B \{coimpl} C(x)"/>
      ...
</aufgabe>
</aufgabenblatt>
```

Abb. 6: XML-Spezifikation der Aufgabe aus Abbildung 2

5.3 Aufgabentyp Wahrheitstabelle

Der `typ`-Wert einer *Wahrheitstabelle*-Aufgabe ist `WTabelle`. Der typspezifische Tag `var` legt eine Variable, die in dem aussagenlogischen Ausdruck verwendet wird, fest, wobei das Attribut `name` den Namen der Variablen angibt. Der Tag `expr` spezifiziert unter dem Attribut `string` einen aussagenlogischen Ausdruck, in dem die zuvor definierten Variablen auftreten. Für diesen Ausdruck generiert Asterix automatisch eine interaktive Wertetabelle, wobei für jede Teilformel

```
<?xml version="1.0"?>
<aufgabenblatt>
<titel>Wahrheitstabelle</titel>
<text>Gegeben sei die aussagenlogische Formel:
      G = ((A \{or} C) \{and} \{not}B)
      \{impl} (\{not}A \{or} (B \{and} C))</text>
<aufgabe typ="WTabelle">
<titel>Teil 1</titel>
<text>Erstellen Sie eine Wahrheitstabelle für G.</text>
<var name="A"/>
<var name="B"/>
<var name=C"/>
<expr string="((A or C) and not B) => (not A or (B and C))"/>
</aufgabe>
...       (Spezifikation für Teil 2)
</aufgabenblatt>
```

Abb. 7: XML-Spezifikation der Aufgabe aus Abbildung 3

und für jede mögliche Belegung der Variablen ein Feld erzeugt wird, in das der Student den Wert dieser Teilformel unter der betreffenden Belegung eingeben kann. In Abb. 7 ist die XML-Spezifikation der Aufgabe aus Abb. 3 angegeben.

5.4 Aufgabentyp Mengenangabe

Der `typ`-Wert einer *Mengenangabe*-Aufgabe ist `Mengen` (vgl. Abb. 8). Mit dem typspezifischen Tag `menge` wird eine Menge definiert, wobei das Attribut `name` den Namen und das Attribut `contents` die Elemente der Menge angibt. Bei dem Tag `eingabe` wird eine einzugebende Menge spezifiziert, wobei das Attribut `name` wiederum den Namen der Menge angibt. Das Attribut `defn` enthält die Definition der Menge mittels der zuvor unter `menge` eingeführten Mengen und unter Verwendung von Mengenoperationen wie kartesisches Produkt, Durchschnitt, Vereinigung oder Differenz. Das Attribut `defnString` gibt an, was als Definition der Menge dem Studenten in der Aufgabenstellung angezeigt werden soll. Aufgrund dieser Darstellung soll der Student die Elemente der Menge bestimmen und in dem dafür vorgesehenen Eingabefeld eingeben. Diese Eingabe wird mit dem unter `defn` spezifizierten Mengenausdruck verglichen, wobei Asterix diesen Mengenausdruck automatisch auswertet. In Abb. 8 ist die XML-Spezifikation der Aufgabe aus Abb. 4 angegeben.

5.5 Aufgabentyp XausN-Tabelle

Der `typ`-Wert einer *XausN-Tabelle*-Aufgabe ist `XausNTabelle`. Mit dem typspezifischen Tag `value` wird ein Wert festgelegt, der Variablen zugeordnet wer-

```
<?xml version="1.0"?>
<aufgabenblatt>
<titel>Aufgabe 1</titel>
<text> Gegeben seien die folgenden Mengen:
       M[sub]1[/sub] := { x \{elem} \{N} | 1 \{let} x \{let} 10
       }
       M[sub]2[/sub] := { a, b, c, d, e }
       A := { 2, 4, 6, 8 } \{subseteq} M[sub]1[/sub]
       B := { 2, 3, 4, 5, 7 } \{subseteq} M[sub]1[/sub]
       C := { a, b } \{subseteq} M[sub]2[/sub]   </text>
<aufgabe typ="Mengen">
<titel>Teil 1</titel>
<text> Geben Sie die Elemente der Mengen E[sub]1[/sub],
       E[sub]2[/sub], E[sub]3[/sub] an:   </text>
<help> A x B besteht aus [i]geordneten Paaren[/i]. [i](a,
       b)[/i]
       ist also etwas anderes als [i](b, a)[/i]. In den
       Mengenklammern spielen dagegen die Reihenfolge und auch
       "Doppelnennungen" keine Rolle: [i]{a, b} = {b, a} =
       {a, a, b}[/i].   </help>
<menge name="A" contents="{2, 4, 6, 8}" />
<menge name="B" contents="{2, 3, 4, 5, 7}" />
<menge name=C" contents="{a, b}"/>
<menge name=M2" contents="{a, b, c, d, e}" />
<eingabe name="E[sub]1[/sub]" defn="(A intersect B) x C"
         defnString="(A \{intersect} B) x C" />
<eingabe name="E[sub]2[/sub]" defn="(A  B) x (M2  C)"
         defnString="(A  B) x (M[sub]2[/sub]  C)" />
<eingabe name="E[sub]3[/sub]" defn="(A intersect C) x B"
         defnString="(A \{intersect} C) x B" />
</aufgabe>
...
</aufgabenblatt>
```

Abb. 8: XML-Spezifikation der Aufgabe aus Abbildung 4

den kann. Das Attribut `id` enthält eine Kurzbezeichnung und das Attribut `name` enthält einen Namen für den Wert. Die Namen der Variablen erscheinen als Zeilenbezeichner, während die Namen der zugehörigen Werte die Spalten bezeichnen. So enthält z.B. die XML-Spezifikation der Aufgabe aus Abb. 5:

```
<aufgabe typ="XausNTabelle">
        ...
     <value id="symm"   name="symmetrisch" />
     <value id="refl"   name="reflexiv" />
     <value id="trans"  name="transitiv" />
     <var richtig="refl"       name="1." />
     <var richtig="symm;refl"  name="2." />
     <var richtig="trans"      name="3." />
</aufgabe>
```

Dabei legt der Tag `var` eine Variable fest, der Werte zugeordnet werden können. Das Attribut `richtig` enthält die durch Semikoli getrennten Kurzbezeichner der korrekten Werte, und das Attribut `name` enthält einen kurzen Text für diese Variable. Hieraus erzeugt Asterix eine strukturierte Darstellung der Antwortmöglichkeiten, wie sie beispielsweise in Abb. 5 wiedergegeben ist.

6 Zusammenfassung und zukünftige Arbeiten

Das Asterix-System enthält interaktive Selbsttestaufgaben unterschiedichen Typs aus dem Bereich der formalen Grundlagen der Informatik, deren Implementierung automatisch aus XML-Spezifikationen generiert wird. Eine mögliche Erweiterung von Asterix ist die Kopplung an ein intelligentes Tutorsystem (ITS) [Lel99]. Mit der internen Verfügbarkeit eines Tutandenmodells könnten die in Asterix vorgesehenen Tipps für die einzelnen Studenten und Fehlersituationen angepasst werden.

Literatur

[BCH+03] O. Borch, S. Crozat, J. Hüvelmeijer, A.-K. Kairamo, R. Lauhia, A. Opsomer, C. Pastrav, O. Rokkjær, N. Sclater, J. Van Den Branden, H. Wedde, and M. Widera. Report cEVU working group 1: Digital Platforms. In *International Conference on Networked e-learning for European Universities*. Europace, November 2003.

[BIKI+05] C. Beierle, M. Isberner, G. Kern-Isberner, B. Messing, and M. Widera. Generierung interaktiver Selbsttestaufgaben im Bereich der formalen Grundlagen der Informatik aus XML-Spezifikationen. In J. Haake, U. Lucke, and D. Tavangarian, editors, *DeLFI 2005: 3.*

e-Learning Fachtagung Informatik, volume P-66 of *Lecture Notes in Informatics (LNI)*. Köllen Verlag, 2005.

[BPSM+04] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. *Extensible Markup Language (XML) 1.0 (Third Edition)*. W3C, February 2004. http://www.w3.org/TR/2004/REC-xml-20040204/.

[Lel99] R. Lelouche. Intelligent tutoring systems from birth to now. *Künstliche Intelligenz*, 13(4):5–11, November 1999.

[LGH02] R. Lütticke, C. Gnörlich, and H. Helbig. Vilab - a virtual electronic laboratory for applied computer science. In *Proceedings of the Conference Networked Learning in a Global Environment*. ICSC Academic Press, Canada/The Netherlands, 2002.

[LVU05] Lernraum Virtuelle Universität, Fernuniversität in Hagen, `http://www.fernuni-hagen.de/LVU/`. 2005.

[WWR05] Wissenswerkstatt Rechensysteme, `http://www.wwr-project.de`. 2005.